


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide


THE ACM DIGITAL LIBRARY

Feedback

 +snapshot* +summary map* +logical OR* +write allocation*
 Terms used: snapshot summary map logical write allocation

Found 47 of 240,237

Sort results by

relevance

Save results to a Binder

 Refine these results with [Advanced Search](#)

Display results

expanded form

☐ Open results in a new window
Try this search in [The ACM Guide](#)

Results 1 - 20 of 47

Result page: 1 2 3 [next](#) [>>](#)

1 Concurrent programming without locks



Keir Fraser, Tim Harris

May 2007 ACM Transactions on Computer Systems (TOCS). Volume 25

Issue 2

Publisher: ACM

Full text available: pdf (1.58 MB) Additional Information: full citation, abstract, references, index terms

Mutual exclusion locks remain the *de facto* mechanism for concurrency control on shared-memory data structures. However, their apparent simplicity is deceptive: It is hard to design scalable locking strategies because locks can harbor problems ...

Keywords: Concurrency, lock-free systems, transactional memory

2 Capsule: an energy-optimized object storage system for memory-constrained sensor devices



Gaurav Mathur, Peter Desnoyers, Deepak Ganesan, Prashant Shenoy

October 2006 SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems

Publisher: ACM

Full text available: pdf (470.09 KB) Additional Information: full citation, abstract, references, cited by, index terms

Recent gains in energy-efficiency of new-generation NAND flash storage have strengthened the case for in-network storage by data-centric sensor network applications. This paper argues that a simple file system abstraction is inadequate for realizing ...

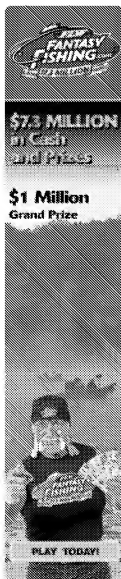
Keywords: embedded systems, energy efficiency, file system, flash memory, objects, sensor network, storage system

3 hFS: a hybrid file system prototype for improving small file and metadata performance




Zhihui Zhang, Kanad Ghose

June 2007 ACM SIGOPS Operating Systems Review. Volume 41 Issue 3


 Fantasy fishing event
 held by Chronicle

Publisher: ACM

Full text available:  pdf(429.39 KB) Additional Information: full citation, abstract, references, index terms

Two oft-cited file systems, the Fast File System (FFS) and the Log-Structured File System (LFS), adopt two sharply different update strategies---*update-in-place* and *update-out-of-place*. This paper introduces the design and implementation ...

Keywords: disk inodes, file systems, metadata journaling, update strategies


4 hFS: a hybrid file system prototype for improving small file and metadata performance



Zhihui Zhang, Kanad Ghose

March 2007 EuroSys '07: Proceedings of the ACM SIGOPS/EuroSys European Conference on Computer Systems 2007

Publisher: ACM

Full text available:  pdf(429.39 KB) Additional Information: full citation, abstract, references, index terms

Two oft-cited file systems, the Fast File System (FFS) and the Log-Structured File System (LFS), adopt two sharply different update strategies---*update-in-place* and *update-out-of-place*. This paper introduces the design and implementation ...

Keywords: disk inodes, file systems, metadata journaling, update strategies

5 GPGPU: general purpose computation on graphics hardware



David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn

August 2004 SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes

Publisher: ACM

Full text available:  pdf(69.03 MB) Additional Information: full citation, abstract, cited by

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex ...


6 The KaffeOS Java runtime system



Godmar Back, Wilson C. Hsieh

July 2005 ACM Transactions on Programming Languages and Systems (TOPLAS). Volume 27 Issue 4

Publisher: ACM

Full text available:  pdf(704.30 KB) Additional Information: full citation, abstract, references, cited by, index terms, review

Single-language runtime systems, in the form of Java virtual machines, are widely deployed platforms for executing untrusted mobile code. These runtimes provide some of the features that operating systems

provide: interapplication memory protection and ...

Keywords: Robustness, garbage collection, isolation, language runtimes, resource management, termination, virtual machines

7 Schedulable persistence system for real-time applications in virtual machine



Okehee Goh, Yann-Hang Lee, Ziad Kaakani

October 2006 EMSOFT '06: Proceedings of the 6th ACM & IEEE International conference on Embedded software

Publisher: ACM

Full text available: pdf(254.72 KB) Additional Information: full citation, abstract, references, index terms

Persistence in applications saves a computation state that can be used to facilitate system recovery upon failures. As we begin to adopt virtual execution environments (VMs) for mission-critical real-time embedded applications, persistence service will ...

Keywords: CLI, checkpoint/recovery, real-time applications, schedulable persistence system, virtual machine

8 Dynamo: amazon's highly available key-value store



Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan

Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan

Sivasubramanian, Peter Voshall, Werner Vogels

October 2007 SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles

Publisher: ACM

Full text available: pdf(894.87 KB) Additional Information: full citation, abstract, references, index terms

Reliability at massive scale is one of the biggest challenges we face at Amazon.com, one of the largest e-commerce operations in the world; even the slightest outage has significant financial consequences and impacts customer trust. The Amazon.com platform, ...

Keywords: performance, reliability, scalability

9 MC2: high-performance garbage collection for memory-constrained environments



Narendran Sachindran, J. Eliot B. Moss, Emery D. Berger

October 2004 ACM SIGPLAN Notices, Volume 39 Issue 10

Publisher: ACM

Full text available: pdf(503.63 KB) Additional Information: full citation, abstract, references, cited by, index terms

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory ...

Keywords: copying collector, generational collector, java, mark-

compact, mark-copy, mark-sweep, memory-constrained copying

10 MC²: high-performance garbage collection for memory-constrained environments



Narendran Sachindran, J. Eliot B. Moss, Emery D. Berger
October 2004 OOPSLA '04: Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications

Publisher: ACM

Additional Information: full citation, abstract,

Full text available: pdf(503.53 KB)

references, cited by, index terms

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory ...

Keywords: copying collector, generational collector, java, mark-compact, mark-copy, mark-sweep, memory-constrained copying

11 Real-time shading



Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost

August 2004 SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes

Publisher: ACM

Full text available: pdf(7.39 MB) Additional Information: full citation, abstract, cited by

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering ...

12 Compile-Time Concurrent Marking Write Barrier Removal

V. Krishna Nandivada, David Detlefs

March 2005 CGO '05: Proceedings of the international symposium on Code generation and optimization

Publisher: IEEE Computer Society

Additional Information: full citation, abstract,

Full text available: pdf(225.35 KB)

references, cited by, index terms

Garbage collectors incorporating concurrent marking to cope with large live data sets and stringent pause time constraints have become common in recent years. The snapshot-at-the-beginning style of concurrent marking has several advantages over the incremental ...

13 Charles W. Bachman interview: September 25-26, 2004; Tucson,



Arizona
Thomas Haigh

January 2006 ACM Oral History interviews

Publisher: ACM

Full text available:  pdf(974.87 KB) Additional Information: full citation, abstract, index terms


Charles W. Bachman reviews his career. Born during 1924 in Kansas, Bachman attended high school in East Lansing, Michigan before joining the Army Anti Aircraft Artillery Corp, with which he spent two years in the Southwest Pacific Theater, during ...

14 EXPLODE: a lightweight, general system for finding serious storage system errors

Junfeng Yang, Can Sar, Dawson Engler


November 2006 OSDI '06: Proceedings of the 7th symposium on Operating systems design and implementation

Publisher: USENIX Association

Full text available:  pdf(731.30 KB) Additional Information: full citation, abstract, references


Storage systems such as file systems, databases, and RAID systems have a simple, basic contract: you give them data, they do not lose or corrupt it. Often they store the only copy, making its irrevocable loss almost arbitrarily bad. Unfortunately, their ...

15 Software behavior oriented parallelization

 Chen Ding, Xipeng Shen, Kirk Kelsey, Chris Tice, Ruke Huang, Chengliang Zhang

June 2007 ACM SIGPLAN Notices, Volume 42 Issue 6


Publisher: ACM

Full text available:  pdf(299.09 KB) Additional Information: full citation, abstract, references, index terms

Many sequential applications are difficult to parallelize because of unpredictable control flow, indirect data access, and input-dependent parallelism. These difficulties led us to build a software system for behavior oriented parallelization (BOP), ...


Keywords: program behavior, speculative parallelization

16 Software behavior oriented parallelization

 Chen Ding, Xipeng Shen, Kirk Kelsey, Chris Tice, Ruke Huang, Chengliang Zhang

June 2007 PLDI '07: Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation

Publisher: ACM

Full text available:  pdf(299.09 KB) Additional Information: full citation, abstract, references, index terms

Many sequential applications are difficult to parallelize because of unpredictable control flow, indirect data access, and input-dependent parallelism. These difficulties led us to build a software system for behavior oriented parallelization (BOP), ...

Keywords: program behavior, speculative parallelization

17 [Language support for lightweight transactions](#)



Tim Harris, Keir Fraser

November 2003 ACM SIGPLAN Notices, Volume 38 Issue 11

Publisher: ACM

Full text available: pdf(224.15 KB)

Additional Information: full citation, abstract,
references, cited by, index
terms

Concurrent programming is notoriously difficult. Current abstractions are intricate and make it hard to design computer systems that are reliable and scalable. We argue that these problems can be addressed by moving to a declarative style of concurrency ...

Keywords: concurrency, conditional critical regions, non-blocking systems, transactions

18 [Language support for lightweight transactions](#)



Tim Harris, Keir Fraser

October 2003 OOPSLA '03: Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications

Publisher: ACM

Full text available: pdf(224.15 KB)

Additional Information: full citation, abstract,
references, cited by, index
terms

Concurrent programming is notoriously difficult. Current abstractions are intricate and make it hard to design computer systems that are reliable and scalable. We argue that these problems can be addressed by moving to a declarative style of concurrency ...

Keywords: concurrency, conditional critical regions, non-blocking systems, transactions

19 [The design and development of ZPL](#)



Lawrence Snyder

June 2007 HOPL III: Proceedings of the third ACM SIGPLAN conference on History of programming languages

Publisher: ACM


Full text available: pdf(2.65 MB)

Additional Information: full citation, appendices and
supplements, abstract,
references, index terms

ZPL is an implicitly parallel programming language, which means all instructions to implement and manage the parallelism are inserted by the compiler. It is the first implicitly parallel language to achieve performance portability, that is, consistent ...

Keywords: CTA, WYSIWYG performance model, parallel language design, performance portability, problem space promotion, regions, type architecture

20 Comparison of access methods for time-evolving data

 Betty Salzberg, Vassilis J. Tsotras

June 1999 ACM Computing Surveys (CSUR), Volume 31 Issue 2

Publisher: ACM

Additional Information: full citation, abstract,

Full text available:  pdf(529.53 KB)references, cited by, index
terms

This paper compares different indexing techniques proposed for supporting efficient access to temporal data. The comparison is based on a collection of important performance criteria, including the space consumed, update processing, and query time for ...

Keywords: I/O performance, access methods, structures, temporal databases

Results 1 - 20 of 47

Result page: 1 2 3 [next](#) >>

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2008 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  Adobe Acrobat  QuickTime  Windows Media Player  Real Player